

Unipolar terminal-attractor-based neural associative memory with adaptive threshold and perfect convergence

Chwan-Hwa Wu and Hua-Kuang Liu

A perfectly convergent unipolar neural associative-memory system based on nonlinear dynamical terminal attractors is presented. With adaptive setting of the threshold values for the dynamic iteration for the unipolar binary neuron states with terminal attractors, perfect convergence is achieved. This achievement and correct retrieval are demonstrated by computer simulation. The simulations are completed (1) by exhaustive tests with all of the possible combinations of stored and test vectors in small-scale networks and (2) by Monte Carlo simulations with randomly generated stored and test vectors in large-scale networks with an M/N ratio of 4 (M is the number of stored vectors; N is the number of neurons ≤ 256). An experiment with exclusive-OR logic operations with liquid-crystal-television spatial light modulators is used to show the feasibility of an optoelectronic implementation of the model. The behavior of terminal attractors in basins of energy space is illustrated by examples.

1. Introduction

One of the major research topics of neural networks is in the area of associative memory. The avalanche of intensive research interest in neural networks was initiated by the research of Hopfield,¹ in which the associative memory is modeled with a neural synaptic interconnection matrix and encompasses an interesting computation scheme that uses recursive, nonlinear thresholding. But further investigation reveals^{2,3} that the storage capacity of the Hopfield model is quite limited because of the number of spurious states and oscillations.

In order to alleviate the spurious-state problems in the Hopfield model, Zak⁴ introduced the concept of terminal attractors. However, the theory of the terminal-attractor-based associated neural-network model proposed by Zak determines that a new synapse matrix totally different from the Hopfield matrix is needed. This new matrix, which is complex and

time consuming to compute, was proven to eliminate spurious states, to increase the speed of convergence, and to control the basin of attraction.^{4,5} Zak's derivation shows that the Hopfield matrix works only if all the stored states in the network are orthogonal. However, since the synapses have changed from those determined by Hebb's law, Zak's model is different from the Hopfield model except for the dynamical iteration of the recall process. The improvement of the storage capacity of the Hopfield model by the terminal attractor cannot be determined based on Zak's model.

More recently, for the purpose of comparing the Hopfield model both including and excluding the terminal attractor, Liu *et al.* proposed a terminal-attractor-based associative-memory (TABAM) model that incorporates binary neurons into the synaptic matrix determined by Hebb's law in the same way as the Hopfield model.⁶ Several techniques for the optical implementation, which include the application of the inner-product approach⁷ and the exclusive-OR (XOR) operation of the liquid-crystal-television spatial light modulator (LCTV SLM),^{8,9} were introduced.

The complexity of the optical implementation of TABAM is discussed in Ref. 6. In general, optical implementation of subtraction increases the complexity of the system; hence a unipolar neuron model, which has 1 and 0 as binary states instead of 1 and -1

C.-H. Wu is with the Department of Electrical Engineering, Auburn University, Auburn, Alabama 36849-5201. H.-K. Liu is with the California Institute of Technology, Pasadena, California 91109-8099.

Received 1 October 1992; revised manuscript received 24 August 1993.

0003-6935/94/112210-08\$06.00/0.

© 1994 Optical Society of America.

(without subtraction or negative numbers), is more suitable. In this paper we present a unipolar inner-product TABAM (UIT)¹⁰ and a cross-talk-reduced inner-product TABAM (CRIT).¹⁰ The latter provides a mechanism to transform the input-state vector into the correct basin of the stored vector; then it uses the terminal attractor to accelerate the convergence. In addition, the UIT and the CRIT models do not require any training time and therefore are ready for optoelectronic implementation. In both the UIT and the CRIT a dynamical logistic function for an adaptive-thresholding technique is developed. An adaptive threshold is determined for every iteration to perform the nonlinear thresholding for the proposed unipolar binary implementation.

Before the optical implementation is designed, it is important first to test, by computer simulation, the validity of the Hopfield matrix, or Hebb's law, in the UIT and the CRIT even though the stored states are not orthogonal. These models need a new algorithm for the determination of the threshold value in the associative recall for a unipolar implementation to reach convergence. The simulations are completed (1) by exhaustive tests with all of the possible combinations of stored and test vectors in small-scale networks and (2) by Monte Carlo simulations with randomly generated stored and test vectors in large-scale networks. The feasibility of an optoelectronic implementation of the technique is also demonstrated with a small number of neurons and stored states. The application of the TABAM for solving pattern-recognition problems is discussed.

In Section 2, the TABAM models are described, and the adaptive-threshold values of the TABAM are determined through theoretical derivations. The computer-simulation algorithm and results are described in Section 3. Both the comparison between the UIT and the CRIT and the behavior of an attractor in an energy basin are described in Section 4. The architecture of optical implementation and demonstrative optical experimental results are described in Section 5. The conclusion is presented in Section 6.

2. Theoretical Discussion

A. Basic Terminal-Attractor-Based Associative-Memory Models

In general, it is difficult to implement subtraction by use of optics. To avoid subtraction, we use an adaptive-thresholding technique in the TABAM models. Before the adaptive-thresholding techniques are discussed, the TABAM⁶ model should be reviewed briefly. In this model one assumes that the i th component of the state vector x_i at time $t + 1$ may be written as a function of t (where t is an integer, a discrete representation of the time steps) as follows:

$$x_i(t + 1) = \sum_{m=1}^M v_i^m \alpha^m(t) + I_i, \quad (1)$$

where

$$I_i = \sum_{m=1}^M \alpha^m(t) \{g[x_i(t)] - v_i^m\}^{1/3} \delta^m(t),$$

$$\delta^m(t) = \exp\left(-\beta \sum_{i=1}^N \{g[x_i(t)] - v_i^m\}^2\right),$$

$$\alpha^m(t) = \sum_{j=1}^N v_j^m g[x_j(t)],$$

where $g[x_j(t)]$ is a logistic function:

$$g[x_j(t)] = 1/(1 + \exp[-a[x_j(t) - \theta(t)]]).$$

The constant a in the logistic (sigmoid, or nonlinear activation) function decides the slope of the nonlinear thresholding function, and $\theta(t)$ is the threshold. In Eq. (1), v_i^m denotes the i th component of the m th stored vector \mathbf{V}^m , β is a constant, and M and N are the number of stored vectors and the number of neurons, respectively. Based on the property of the unipolar representation of a binary system, this equation can be rewritten as⁶

$$x_i(t + 1) = \sum_{m=1}^M \alpha^m(t) \{[1 + \delta^m(t)]v_i^m - g[x_i(t)]\delta^m(t)\}, \quad (2)$$

since

$$g[x_i(t)] - v_i^m = 1, 0, \text{ or } -1,$$

$$\{g[x_i(t)] - v_i^m\}^{1/3} = g[x_i(t)] - v_i^m. \quad (3)$$

B. Unipolar Inner-Product Terminal-Attractor-Based Associative-Memory Model

Instead of implementing the subtraction in Eq. (2) optically, an adaptive-threshold function is introduced following the modification of Eq. (2) as follows:

$$x_i(t + 1) = \sum_{m=1}^M \alpha^m(t) \{[1 + \delta^m(t)]v_i^m + g[x_i(t)]\delta^m(t)\}. \quad (4)$$

We refer to Eq. (4) as the UIT.

In Eq. (4) a logistic function, as illustrated in Fig. 1, with an adaptive threshold, $\theta(t)$, is used to perform the nonlinear transform such that the output attains quasi-binary states, which are defined when the constant a in the logistic function becomes a **very** large number but is still finite. The logistic function is continuous and has a large slope near the threshold, $\theta(t)$. This provides the terminal attractor with a continuous logistic function in a binary system. The selection of the constant a depends on the numerical solution in the system. The adaptive threshold is determined from an ideal case and is then generalized for a realistic case. In the ideal case all of the stored states are orthogonal to one another; hence the system is cross-talk-free.

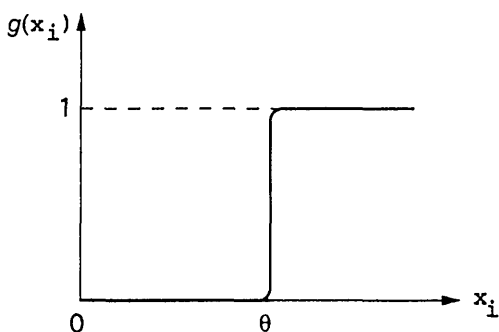


Fig. 1. Dynamic logistic function with a sharp transition at $x = 0$ (threshold).

When there is no cross talk, the threshold $\theta(t)$ is set as

$$\theta(t) = \alpha^{m'}(t)[1/2 + \delta^{m'}(t)], \quad (5)$$

where $\alpha^{m'}(t)$ is the inner product between $x(t)$ and the m th stored vector, $\mathbf{V}^{m'}$, and

$$\alpha^m(t) = 0 \quad \text{for all } m \neq m'.$$

Furthermore, the threshold $\theta(t)$ is set according to the following four possible cases:

(case 1) If $v_i^{m'} = 1$ and $g[x_i(t)] \approx 1$,

$$\text{then } x_i(t+1) \approx \alpha^{m'}(t)[1 + 2\delta^{m'}(t)].$$

(case 2) If $v_i^{m'} = 0$ and $g[x_i(t)] \approx 0$,

$$\text{then } x_i(t+1) \approx 0.$$

(case 3) If $v_i^{m'} = 1$ and $g[x_i(t)] \approx 0$,

$$\text{then } x_i(t+1) \approx \alpha^{m'}(t)[1 + \delta^{m'}(t)].$$

(case 4) If $v_i^{m'} = 0$ and $g[x_i(t)] \approx 1$,

$$\text{then } x_i(t+1) \approx \alpha^{m'}(t)\delta^{m'}(t). \quad (6)$$

In order to cause the states to converge to the stored states, one should set a threshold on the basis of Eq. (5) for maximum noise immunity between cases (3) and (4), as shown in Fig. 2. The threshold is set between the lower value of the desired output

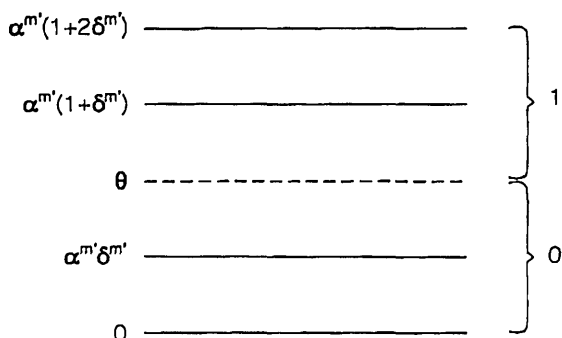


Fig. 2. Adaptive threshold, θ , is set to achieve the maximum noise immunity.

(state 1) and the higher value of the desired output (state 0).

For a more practical situation the stored states are not orthogonal to one another. In this situation, cross talk often occurs between similar stored vectors. For the nonorthogonal case, $\theta(t)$ is set to be

$$\theta(t) = \sum_{m=1}^M \alpha^m(t)[(1/2) + \delta^m(t)]. \quad (7)$$

C. Cross-Talk-Reduced Inner-Product Terminal-Attractor-Based Associative-Memory Model

When the cross talk corrupts the desired state because of the weighing process of the inner-product approach and because every stored vector contributes the inner product to the summation in Eq. (1), even though each stored vector is quite different from the input vector (as long as the inner product is not zero), the threshold $\theta(t)$ set by Eq. (7) cannot provide an appropriate thresholding mechanism. In order to reduce or to eliminate the cross-talk problem, we propose a new model called CRIT:

$$x_i(t+1) = \sum_{m=1}^M \alpha^m(t)\delta^m(t)[[1 + \delta^m(t)]v_i^m + g[x_i(t)]\delta^m(t)]. \quad (8)$$

Equation (8) indicates that the cross talk between the nonorthogonal stored vectors is exponentially weighed and reduced by the exponential term $\delta^m(t)$. The property of the basin of terminal attractors is controlled by the value of β in $\delta^m(t)$, which is discussed in detail in Section 4. With the model the retrieved state vector is highly likely to be placed in a correct basin, similar to the effect resulting from the multiplication between the weight matrix and the nonlinearly activated input vector in Zak's model. Then the corresponding terminal attractor forces the state vector to converge rapidly to the bottom of the basin. This model does not need time for training and is more suitable for optoelectronic implementation for terminal attractors since the stored vectors can be used directly. The threshold can be set in a manner similar to that used in Eq. (7), as follows:

$$\theta(t) = \sum_{m=1}^M \alpha^m(t)\delta^m(t)[(1/2) + \delta^m(t)]. \quad (9)$$

3. Computer Simulation

A. Simulation Algorithm

In order to test the effectiveness of the models of UIT and CRIT with the adaptive threshold, we use a computer simulation. The XOR logic operation is used to detect the Hamming distance between the state vector, $x(t)$, and the stored vectors, \mathbf{V}^m . The input-output relationship of XOR operation is shown in Table 1. The Hamming distance between the state vector, $x(t)$, and the stored vectors can be

Table 1. Exclusive-OR (XOR) Relationship

Input 1	Input 2	Output
1	1	1
0	1	0
1	0	0
0	0	1

computed by use of XOR as follows. For a unipolar system,

$$\{g[x_i(t)] - v_i^m\}^2 \approx |g[x_i(t)] - v_i^m|; \quad (10)$$

because $g[x_i(t)]$ is very close to 0 or 1 and v_i^m is 0 or 1, Eq. (10) is a reasonable approximation. The Hamming distance between $g[x_i(t)]$ and v_i^m is shown in Table 2.

Based on the Tables 1 and 2, the Hamming distance is found to be

$$\begin{aligned} \sum_{i=1}^N \{g[x_i(t)] - v_i^m\}^2 &= \sum_{i=1}^N |g[x_i(t)] - v_i^m| \\ &= \sum_{i=1}^N \{1 + g[x_i(t)] \text{ XOR } v_i^m\} \\ &= N - \alpha^m(t), \end{aligned} \quad (11)$$

where

$$\alpha^m(t) = \sum_{i=1}^N g[x_i(t)] \text{ XOR } v_i^m.$$

Equation (11) provides a similar measure between $g[x(t)]$ and \mathbf{V}^m and is equivalent to the inner product of a bipolar system. The exponential term can be calculated easily after the Hamming distance is obtained:

$$\delta^m(t) = \exp\{-\beta[N - \alpha^m(t)]\}, \quad (12)$$

based on the definition in Eq. (1).

B. Simulation Results

Based on the above algorithm, two computer-simulation programs are used to test the feasibility of the models: (1) an exhaustive test program for simulating every possible stored and input vector in small-scale networks of UIT and CRIT and (2) a Monte Carlo simulation code for testing randomly generated stored and input vectors in large-scale networks of CRIT.

Table 2. Hamming Distance between $g[x_i(t)]$ and v_i^m

$g[x_i(t)]$	v_i^m	$ g[x_i(t)] - v_i^m $
1	1	0
0	1	1
1	0	1
0	0	0

In the first program the number of neurons and stored states selected is small. The number of stored vectors is two and three for all the possible states in both three- and four-neuron networks. For small N and M all possible combinations of vectors are stored and retrieval is tested for all possible input vectors by use of an IBM PC 386 for the simulation. An associative-recall result is considered accurate when the Hamming distance between the input vector and the stored vector to which it converges is the smallest. The accuracy can be determined in all cases except those in which the input vector has a Hamming distance equal to two or more stored vectors. In this case it is impossible to decide to which of the stored states the input should converge; however, because the Hamming distances are equal, the choice is inconsequential. The simulation results of UIT [Eq. (4)] and the CRIT [Eq. (7)] are presented in Table 3. The 95.7% convergence accuracy of $M = 3$ and $N = 4$ of the UIT is because of the cross-talk effect. The inaccuracy is removed by the CRIT, as shown in Table 3. When M and N are further increased, the computer time for exhaustive testing simulation becomes considerably long. Hence a Monte Carlo simulation is used to test randomly generated patterns to evaluate the accuracy of retrieval.

For the second program a Monte Carlo simulation code is developed in a SunSPARC 2 workstation to simulate a CRIT network with a set of randomly generated stored vectors (all of which are different) and to test the network with a set of randomly generated input vectors to measure the accuracy of correct convergence. As shown in Table 4, only 16 cases are selected from a significant number of performed tests to demonstrate the perfect convergence of the CRIT network. Each row in the table specifies the number of neurons in the network, the number of stored vectors in a set, the number of test vectors in a set, and the number of sets of stored vectors tested. For example, the last row of Table 4 is described as follows: We use 64 different sets of stored patterns for testing. Each set of stored patterns has 1024 patterns and is tested by a set of 256 test patterns. Each set of test patterns is generated randomly and independently for each set of stored patterns. Based on the Monte Carlo simulation results, the CRIT

Table 3. Computer Simulation Results of Unipolar Inner-Product and Cross-Talk-Reduced Terminal-Attractor-Based Associative-Memory (UIT) and (CRIT) Models

Model	N	M	Convergence (%)
UIT	3	2	100
UIT	3	3	100
UIT	4	2	100
UIT	4	3	95.7
CRIT	3	2	100
CRIT	3	3	100
CRIT	4	2	100
CRIT	4	3	100

Table 4. Monte Carlo Simulation Results of the Cross-Talk-Reduced Terminal-Attractor-Based Associative-Memory Model in Large-Scale Networks^a

Number of Neurons	Number of Stored Patterns	Number of Test Patterns	Number of Sets of Stored Patterns
4	8	16	16
9	18	64	64
9	36	64	64
16	32	64	64
16	64	64	64
25	50	64	64
25	100	64	64
36	72	64	64
36	144	64	64
64	128	64	64
64	256	64	64
128	256	128	64
128	512	128	64
256	512	256	64
256	1024	256	64

^aConvergence has 100% accuracy in all 16 cases.

model works well, achieving 100% accuracy, for large-scale networks even when $M = 4N$.

Based on the results of the computer simulation, it can be seen that even with $M = 4N$, perfectly accurate convergence can be accomplished by CRIT. A more detailed comparison between the UIT and by CRIT models is discussed in Section 4. The details of a typical example are provided together with the optoelectronic implementation of the models in Section 5.

4. Comparison between Unipolar Inner-Product and Cross-Talk-Reduced Terminal-Attraction-Based Associative-Memory Models

The difference between the UIT and the CRIT schemes is illustrated as follows. The following three vectors, which are stored in a four-neuron network,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

are used to exemplify the properties of the UIT and the CRIT in this section. Since there are four binary neurons in the network, a thorough test of all 16 possible patterns as the input to the UIT and the CRIT schemes is tabulated in Table 4. The precision of convergence of each test pattern is judged by the closeness of the Hamming distance between the output state and a stored vector. When the distance is a minimum, the stored vector is a nearest neighbor. If any of the two or more stored vectors have an identical Hamming distance to the test pattern, then the retrieval is inconsequential. For example, test pattern 8,

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix},$$

has a two-bit Hamming distance to stored patterns 1 and 3; hence the outputs from the UIT and the CRIT schemes (in Table 5) are considered to be correct because they are the summations of the weighed stored vectors. The result of the UIT is different from that of the CRIT since the weighing factors of three stored vectors are different in these two schemes.

There are two errors made by the UIT scheme in Table 5. The analysis of the fault resulting from test pattern 6 with the UIT scheme is performed as follows. The result from the UIT before thresholding is

$$\begin{bmatrix} 2.3 & 4.1 \\ 5.7 & 8.0 \end{bmatrix},$$

where the threshold θ is 5.1 ($\beta = 1.0$). This thresholding causes the output state to remain as

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

forever. Based on Eq. (11),

$$\alpha^1 = 2, \quad \alpha^2 = 3, \quad \alpha^3 = 2.$$

The cross talk from patterns 1 and 3 contributes weighting factor 2 ($\alpha^1 = 2$ and $\alpha^3 = 2$) each, whereas pattern 2 contributes weighting factor 3 ($\alpha^2 = 3$). The next state of Eq. (3) is the summation of the weighting process, in which the effect of α^1 and α^3 corrupts the second neuron to state 0 in contrast to the weighting factor α^2 , since it does not have enough strength to pull the state of the second neuron to

Table 5. Comparison of the Neural Associative-Retrieval Outputs of the Unipolar Inner-Product and the Cross-Talk-Reduced Terminal-Attractor-Based Associative-Memory (UIT) and (CRIT) Models for Three Stored Patterns (1001, 0111, 0000)

Pattern Number	Test Pattern	UIT Output	UIT Correctness	CRIT Output	CRIT Correctness
1	1 1 1 1	0 1 1 1	Yes	0 1 1 1	Yes
2	0 1 1 1	0 1 1 1	Yes	0 1 1 1	Yes
3	1 0 1 1	1 0 0 1	Yes	1 0 0 1	Yes
4	1 1 0 1	1 0 0 1	Yes	1 0 0 1	Yes
5	1 1 1 0	0 1 1 1	Yes	0 1 1 1	Yes
6	0 0 1 1	0 0 1 1	No	0 1 1 1	Yes
7	1 0 0 1	1 0 0 1	Yes	1 0 0 1	Yes
8	1 1 0 0	0 0 0 1	Yes ^a	1 0 0 0	Yes ^a
9	0 1 1 0	0 1 1 1	Yes	0 1 1 1	Yes
10	1 0 1 0	0 0 0 1	Yes ^a	1 0 0 0	Yes ^a
11	0 1 0 1	0 1 0 1	No	0 1 1 1	Yes
12	0 0 0 0	0 0 0 0	Yes	0 0 0 0	Yes
13	1 0 0 0	1 0 0 0	Yes ^a	1 0 0 0	Yes ^a
14	0 1 0 0	0 0 0 0	Yes	0 0 0 0	Yes
15	0 0 1 0	0 0 0 0	Yes	0 0 0 0	Yes
16	0 0 0 1	0 0 0 1	Yes ^a	0 0 0 1	Yes ^a

^aThe test pattern has the same Hamming distance to at least two of the stored patterns.

state 1. In addition,

$$\delta^1 = e^{-2}, \quad \delta^2 = e^{-1}, \quad \delta^3 = e^{-2},$$

such that these δ values cannot improve the threshold θ set by Eq. (7) to generate a value appropriate for distinguishing the states 0 and 1. Because the cross talk in the inner-product approach is large enough to corrupt the correct state, an error is made.

The CRIT uses the δ^m to improve the weighting process so that the signal-to-noise ratio is better for the thresholding process since the weighting factor decreases exponentially as the Hamming distance increases. When the β is set to 1, the output of the CRIT before thresholding is

$$\begin{bmatrix} 0.3 & 1.5 \\ 2.0 & 2.3 \end{bmatrix},$$

where the threshold θ is 1.3. After taking the thresholding, we obtain the result:

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},$$

which is the correct state, and this state remains the same during the iterative process. In order to illustrate the effect of the values of β , we now show the thresholding processes for β values of 2 and 3. The output of the CRIT before thresholding is

$$\begin{bmatrix} 0.0 & 0.5 \\ 0.5 & 0.6 \end{bmatrix} \quad (13)$$

when $\beta = 2$. The threshold θ is 0.3, and the CRIT

converges to a correct state after thresholding. In comparison with the previous case ($\beta = 1$), the cross talk is reduced for neuron 2. The case of $\beta = 3$, the output before thresholding, is

$$\begin{bmatrix} 0.0 & 0.2 \\ 0.2 & 0.2 \end{bmatrix},$$

and the threshold θ is 0.1. Comparing the three cases of β values of 1, 2, and 3 reveals that the cross talk between the three stored patterns is smaller when β is large; hence the threshold setting is clearer when a larger β is used.

The β in Eq. (1) controls the influence that a stored pattern has on its neighbor (in terms of Hamming distance). A small β causes stronger cross talk between stored patterns, whereas a large β reduces the cross talk. In general, the values of β decide the behavior of neighbors of a terminal attractor. In other words, a pattern has a clear boundary of a basin when a large β is used in the CRIT. On the contrary, spurious states caused by cross talk are generated on the boundary of a basin of a stored pattern when the UIT or the CRIT with a small β are used.

5. Optoelectronic Implementation: a Feasibility Demonstration

A. Experimental Architecture

In discussing the feasibility of optoelectronic implementation of the models, we describe a typical example with four neurons and two stored states. The experimental architecture is shown in Fig. 3. The LCTV's from an Epson video projector are used as the SLM's. The input from a frame grabber of an IBM

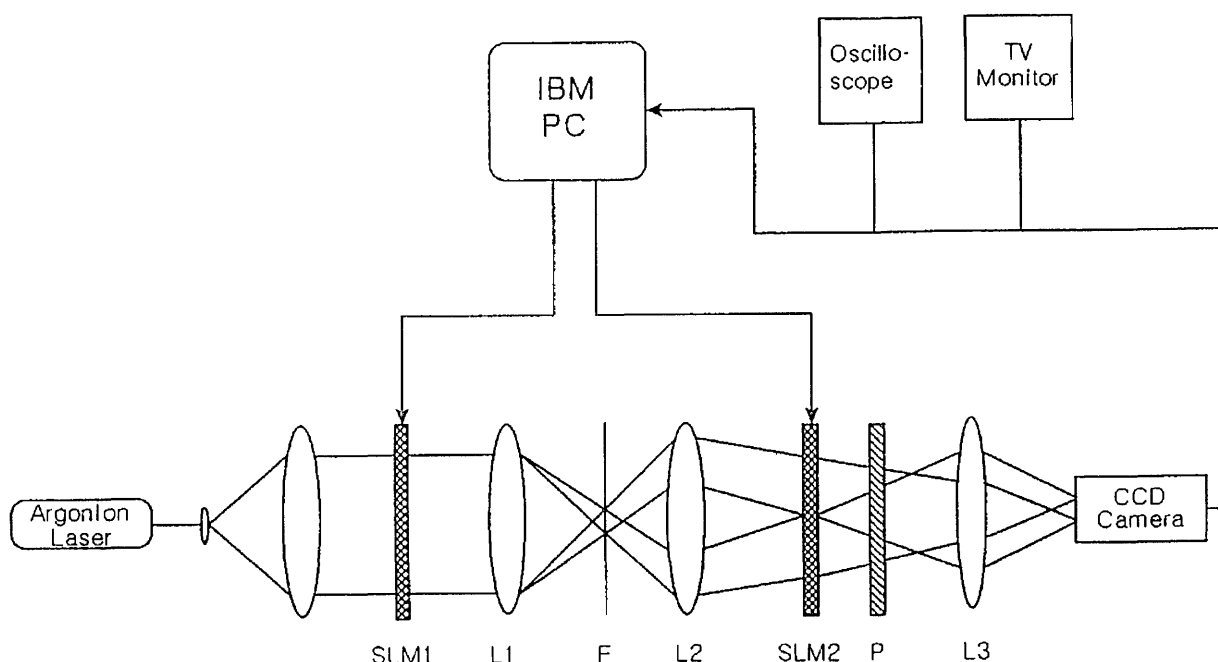


Fig. 3. Experimental setup for the optical XOR operations.

PC computer is sent to SLM1, which is illuminated by a collimated argon-ion laser beam. The image is Fourier transformed by lens L1 and is filtered by two pinholes in order to pass the +1 and the -1 diffraction orders of grid patterns of the LCTV. The +1 and the -1 orders, once they pass through the pinholes (filters), are inversely Fourier transformed by L2, and the XOR operations are performed with the two images stored in SLM2. The signals resulting from XOR operations are detected by a CCD camera and are sent to the computer to calculate the next state vector, $x(t+1)$. The output $x(t+1)$ is the input as a new state to SLM1 for the dynamical system. This process is iterated until convergence is reached. The converged state is displayed on SLM1 or on a monitor. A more generic example of the inner-product architecture can be found in Ref. 7.

B. Experimental Results

A typical example associated with the experiment is presented below. Assume the input vector is

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

where 1 represents a bright pixel and zero represents a dark pixel. The two stored images (vectors) are assumed to be

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

denoted as stored vectors 1 and 2, respectively. The output images resulting from the XOR operations between the input and the two stored images are

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The output images following the XOR operations appear as two spots on the CCD camera. The intensity ratio of the two spots is 9:4. The experimental result showing the calculated intensity ratio is displayed by an oscilloscope and is shown in Fig. 4. The computer can use an adaptive-thresholding technique as shown in Eqs. (4) or (8) to perform the iterative computation.

The result obtained for the UIT before the threshold is taken is

$$\begin{bmatrix} 7.7 & 6.4 \\ 5.5 & 7.7 \end{bmatrix},$$

and the threshold, θ , of the logistic function is 3.9

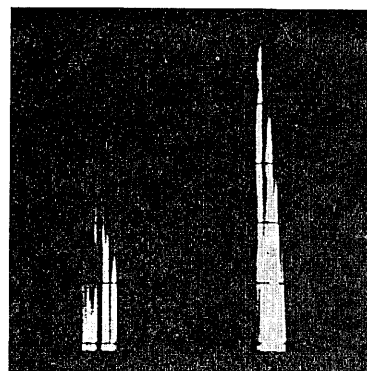


Fig. 4. Oscilloscope display of the intensities of the two spots detected by the CCD camera.

(when one sets $\beta = 1.0$). After the threshold is taken, the result is the stored vector 2, which shows an accurate retrieval. The result of the subsequent iteration before thresholding is

$$\begin{bmatrix} 17 & 17 \\ 13 & 17 \end{bmatrix},$$

where $\theta = 9$. The result after thresholding is again stored vector 2, indicating accurate convergence.

On the other hand, the result obtained for the CRIT before thresholding is

$$\begin{bmatrix} 2.3 & 1.8 \\ 2.0 & 2.3 \end{bmatrix}.$$

After the threshold is taken with $\theta = 1.1$ ($\beta = 1.0$), the result is the same as that of the UIT. The result after the next iteration before thresholding is

$$\begin{bmatrix} 14 & 14 \\ 12 & 14 \end{bmatrix},$$

which converges to the correct state.

The above example offers a feasibility demonstration of use of a computer to perform nonlinear thresholding and feedback. An example of an optical thresholding and feedback is illustrated in Ref. 6, but the fact that a suitable thresholding device is not available limits the capability of purely optical implementation. A hybrid optoelectronic device can be developed to enhance the computation speed of thresholding in a parallel manner.

6. Conclusions

In this paper two unipolar TABAM models, the UIT and the CRIT, are presented with adaptive logistic functions. Computer simulations for the associative retrieval of these models with a small number of neurons and a small number of stored states demonstrate perfect recall and convergence. Corresponding experiments that use the XOR operation of LCTV SLM's demonstrate the feasibility of optoelectronic implementation of the models. The effect of the

values of β in both schemes has been demonstrated through examples. When a large β is used, the CRIT has a clear boundary of a basin of a stored pattern, which behaves as a terminal attractor; however, when a small β is used, spurious states are generated near the boundary of a basin of a stored pattern that does not have an appropriate energy basin for a terminal attractor to attract the input vector.

The perfect convergence for the case of $M = 4N$, even for large-scale networks of CRIT, indicates that the terminal attractors are unique and effective in making significant improvements on the Hopfield model. The proposed model uses an inner-product approach similar to the Hopfield model; however, the adaptive-thresholding technique takes more time in simulation or more hardware in implementation.

In a pattern-recognition/classification problem, the number of pixels of the input image is usually of the order of 10^2 to 10^3 . The proposed CRIT model requires that the inner product be accurately measured in an optical system. This may not be practical with current technology. In order to solve the problem we conjecture that large-dimension neurons may be divided into smaller cells. The data may therefore be reduced by cascading of the processors (basic building blocks, or modules, consisting of a small number of neurons) according to the rules of the CRIT. At each stage the accuracy of convergence is perfect. This paper assesses the accuracy of data retrieval of each module. Even a small four-neuron module can be used as a building block to form a multistage large-scale network. The potentials of these models, such as the cellular approach and pyramidal multiresolution image classification based on UIT and CRIT, are under investigation.

This paper describes research that was jointly sponsored by the Advanced Research Projects Agency and the Information Systems Branch, Office of Space Science Applications, NASA, and was performed by

the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. C. Wu was supported by the Center of the Commercial Development for Space Power and Advanced Electronics/NASA. We thank Joseph Diep for his experimental assistance, Annie Aroyan for typing the manuscript, and Elizabeth A. Johnson, Margaret Hartley, and Regina Dragoin for editing this manuscript.

References

1. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. (USA)* **79**, 2254-2258 (1982).
2. R. J. McElieve, E. C. Posner, E. R. Rodemich, and S. S. Venkatesch, "The capacity of the Hopfield associative memory," *IEEE Trans. Inf. Theory* **IT-33**, 461-482 (1987).
3. B. L. Montgomery and B. V. K. Vijaya Kumar, "Evaluation of the use of Hopfield neural network model as a nearest neighbor algorithm," *Appl. Opt.* **25**, 3759-3766 (1986).
4. M. Zak, "Terminal attractors for addressable memory in neural networks," *Phys. Lett. A* **133**, 18-22 (1988).
5. M. Zak, "Terminal attractors in neural networks," *Neural Networks* **2**, 259-274 (1989).
6. H. K. Liu, J. Barhen, and N. H. Farhat, "Optical implementation of terminal attractor based associative memory," *Appl. Opt.* **31**, 4631-4644 (1992).
7. S. Y. Kung and H. K. Liu, "An optical inner-product array processor for associative retrieval," in *Nonlinear Optics and Applications*, P. A. Yeh, ed., *Proc. Soc. Photo-Opt. Instrum. Eng.* **613**, 214-219 (1986).
8. H. K. Liu and T. H. Chao, "Liquid crystal television spatial light modulators," *Appl. Opt.* **28**, 4772-4780 (1989).
9. T. Lu, S. Wu, X. Wu, and F. T. S. Yu, "Optical implementation of programmable neural networks," in *Optical Pattern Recognition*, H. K. Liu, ed., *Proc. Soc. Photo-Opt. Instrum. Eng.* **1053**, 30-39 (1989).
10. C.-H. Wu and H. K. Liu, "A unipolar terminal-attractor based neural associative memory with adaptive threshold and perfect convergence," in *Proceedings of International Joint Conference on Neural Networks* (Institute of Electrical and Electronics Engineers, New York, 1992), Vol. 1, pp. 47-52.